REMARKS

Applicants are in receipt of the Office Action mailed February 12, 2004. Claims 1-66 are currently pending. Reconsideration is respectfully requested in light of the following remarks.

On p. 2 of the Office Action, the Examiner stated: "Applicants are required to update the Serial number or Application number mentioned in the specification (see page 1 of specification)." Applicants note that the Serial numbers of the applications listed on page 1 of the specification are all correct.

Section 103(a) Rejections:

The Office Action rejected claims 1-7, 10, 11, 13, 14, 16, 17, 19-21, 24, 25, 27-29, 33-39, 44-46, 48, 49, 51-55, 58, 59, 61-63 and 65 under 35 U.S.C. § 103(a) as being unpatentable over Buckle (UK Patent 2,332,288 A) in view of Jagannathan (U.S Patent 6,496,871), claims 8, 9, 41-43, 56 and 57 in further view of Graham (U.S. Patent 6,594,700), claims 12, 15, 22, 23, 47 and 64 in further view of Edward ("Core Jini" pages 405-410), claims 18, 26, 40, 50, 60 and 66 in further view of Emmerich ("Incremental Code Mobility with XML" pages 1-10), claims 30-32 in further view of the "Official Notice". Applicants respectfully traverse these rejections in light of the following remarks.

In regard to claim 1, contrary to the Examiner's assertion, the cited art does not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and storing the data representation language representation of the current computation state of the process. Instead, Buckle discloses a method in which "it is possible to transmit binary files or byte code between agents. The byte code sent in a message may include byte code of an agent itself. The agent enabling layer utilizes the CORBA Externalization service to record object and object states (ie an agent) as a stream of data." (Buckle, page

38, lines 12-14). Buckle thus teaches that the byte <u>code</u> of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2. Buckle further discloses "the agent is received by a receiving agent at location L2. The agent code, carried by the current parameter is stored locally at the physical resource at location L2, so that the agent can operate at physical location L2." (Buckle, page 39, lines 6-8.) Buckle thus teaches that the agent code from the message is stored locally on L2 once the message is received on L2.

Nowhere does Buckle teach or suggest *converting* the current computation state of a process into a *data representation language representation* of the current computation state. Buckle does not teach or suggest converting the current computation state of the agent, the byte code of the agent, or the objects and object states of the agent to a data representation language representation.

In addition, nowhere does Buckle teach or suggest storing a data representation language representation of the current computation state of the process. Buckle describes storing the agent <u>code</u> on the receiving location of the message so that the agent can operate at the physical location (L2). Buckle does not teach or suggest storing the current <u>computation state</u> of the agent, the byte code of the agent, or the objects and object states of the agent independent of receipt of the message at another location.

The Examiner further cites Jagannathan, asserting that Jagannathan teaches "wherein the process comprises information about the execution (process execution, col. 6 lines 40-45) of the process." Jagannathan teaches that "[m]igration of such agents, even during process execution, is straightforward and maintains consistency across the network." (Jagannathan, col. 6, lines 41-43). Jagannathan teaches the migration of agents from one machine to another machine, even during process execution. (Jagannathan, column 17, line 50 - column 18, line 23). Specifically, to migrate an agent, Jagannathan teaches: "A Base A task executer is instructed to execute an agent migrate method on the agent comprising Subagent A to migrate Subagent A to Base B on Machine B. The Base A task executer requests a Base A agent manager to obtain agent control data for

Subagent A and send it to a Machine A communication system. The agent control data comprises header information about the migrating agent, along with its tasks and objects. Next, the A Base A task executer requests a Base A task serializer to serialize task objects within Subagent A in task memory, and the Base A task serializer sends the serialized tasks to the Machine A communication system. Similarly, objects are also serialized and sent to the Machine A communication system by the Base A object manager and object serializer...the Machine A communication system then sends the serialized objects, serialized tasks and agent control data for Subagent A over the network to the communication system for Machine B." (Jagannathan, column 17, lines 53-66). Jagannathan further teaches: "After Subagent A' is thus instantiated on Base B, a class request is sent from Base B to Base A over the network...Base A responds to the class request by sending over the network to Base B class files for the agent which are necessary for resuming the agent on Base B. After all migration steps are finished, the memory block for the Subagent A on Base A is released, and the agent resumes as Subagent A' on the Base B..." (Jagannathan, column 18, lines 12-19). Jagannathan thus teaches that, to migrate an agent, agent control data is sent to the new machine, the objects and tasks of that agent are *serialized* and sent to the new machine, and the class files for the agent are sent to the new machine.

Nowhere does Jagannathan teach or suggest *converting* a current computation state of a process into a *data representation language representation* of the current computation state, nor does Jagannathan teach or suggest *storing* the data representation language representation of the current computation state of the process.

Not only does the cited art (Buckle and Jagannathan) not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and storing the data representation language representation of the current computation state of the process, there is no teaching, suggestion, or motivation to combine the art to teach these limitations. Buckle teaches that the byte code of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2, and then may

be stored locally on L2. There is no motivation within Buckle to convert the current computation state of a process into a data representation language representation of the current computation state or to store the data representation language representation of the current computation state independent of transmitting the data representation language representation to another location. Jagannathan teaches the migration of agents from one machine to another machine, even during process execution, using serialization. There is no motivation within Jagannathan to convert the current computation state of a process into a data representation language representation of the current computation state or to store the data representation language representation of the current computation state.

Therefore, the rejection of claim 1 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to independent claim 51.

In regard to claim 19, contrary to the Examiner's assertion, the cited art does not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and sending the data representation language representation of the current computation state of the process to a second device. Instead, Buckle discloses a method in which "it is possible to transmit binary files or byte code between agents. The byte code sent in a message may include byte code of an agent itself. The agent enabling layer utilizes the CORBA Externalization service to record object and object states (ie an agent) as a stream of data." (Buckle, page 38, lines 12-14.) Buckle thus teaches that the byte <u>code</u> of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2 as a "stream of data."

Nowhere does Buckle teach or suggest *converting* the current computation state of a process into a *data representation language representation* of the current computation state and sending the data representation language representation of the current computation state of the process to a second device. Buckle teaches including

byte code of an agent in a message, and recording object and object states (i.e. an agent) as a *stream of data*. Buckle does not teach or suggest *converting* the current computation state of the agent, the byte code of the agent, or the objects and object states of the agent to a *data representation language representation* for sending to a second device.

Jagannathan teaches the migration of agents from one machine to another machine, even during process execution using serialization. (Jagannathan, column 17, line 50 - column 18, line 23). Nowhere does Jagannathan teach or suggest *converting* a current computation state of a process into a *data representation language representation* of the current computation state and sending the data representation language representation of the current computation state of the process to a second device.

Not only does the cited art (Buckle and Jagannathan) not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and sending the data representation language representation of the current computation state of the process to a second device, there is no teaching, suggestion, or motivation to combine the art to teach these limitations. Buckle teaches that the byte code of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2. There is no motivation within Buckle to convert the current computation state of a process into a data representation language representation of the current computation state for sending the data representation language representation of the current computation state of the process to a second device. Jagannathan teaches the migration of agents from one machine to another machine, even during process execution, using serialization. There is no motivation within Jagannathan to convert the current computation state of a process into a data representation language representation of the current computation state for sending the data representation language representation of the current computation state of the process to a second device.

Therefore, the rejection of claim 19 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested. Similar arguments apply in regard to independent claims 44 and 61.

In regard to claim 27, contrary to the Examiner's assertion, the cited art does not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and sending the data representation language representation of the current computation state of the process to a space service for storage. Instead, Buckle discloses a method in which "it is possible to transmit binary files or byte code between agents. The byte code sent in a message may include byte code of an agent itself. The agent enabling layer utilizes the CORBA Externalization service to record object and object states (ie an agent) as a stream of data." (Buckle, page 38, lines 12-14.) Buckle thus teaches that the byte <u>code</u> of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2. Buckle further discloses "the agent is received by a receiving agent at location L2. The agent code, carried by the current parameter is stored locally at the physical resource at location L2, so that the agent can operate at physical location L2." (Buckle, page 39, lines 6-8.) Buckle thus teaches that the agent code from the message is stored locally on L2 once the message is received on L2.

Nowhere does Buckle teach or suggest *converting* the current computation state of a process into a *data representation language representation* of the current <u>computation state</u>. In addition, nowhere does Buckle teach or suggest sending the *data representation language representation* of the current computation state <u>to a space</u> <u>service</u> for storage. Buckle teaches including byte code of an agent in a message, and recording object and object states (i.e. an agent) as a *stream of data*, and storing the agent code on the receiving location of the message so that the agent can operate at the physical location (L2). Buckle does not teach or suggest *converting* the current computation state of the agent, the byte code of the agent, or the objects and object states of the agent to a *data representation language representation* or sending the current computation state of

the agent, the byte code of the agent, or the objects and object states of the agent to a space service for storage.

Jagannathan teaches the migration of agents from one machine to another machine, even during process execution using serialization. (Jagannathan, column 17, line 50 - column 18, line 23). Nowhere does Jagannathan teach or suggest *converting* a <u>current computation state</u> of a process into a *data representation language representation* of the current computation state and sending the data representation language representation of the current computation state of the process <u>to a space service</u> for storage.

Not only does the cited art (Buckle and Jagannathan) not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state and sending the data representation language representation of the current computation state of the process to a space service for storage, there is no teaching, suggestion, or motivation to combine the art to teach these limitations. Buckle teaches that the byte code of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2, and storing the agent code on the receiving location of the message so that the agent can operate at the physical location (L2). There is no motivation within Buckle to convert the current computation state of a process into a data representation language representation of the current computation state for sending the data representation language representation of the current computation state of the process to a space service for storage. Jagannathan teaches the migration of agents from one machine to another machine, even during process execution, using serialization. There is no motivation within Jagannathan to convert the current computation state of a process into a data representation language representation of the current computation state for sending the data representation language representation of the current computation state of the process to a space service for storage.

Therefore, the rejection of claim 27 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested.

In regard to claim 41, similar arguments as discussed above for claim 1 apply. Furthermore, contrary to the Examiner's assertion, the cited art does not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state, storing the data representation language representation of the current computation state of the process, and generating an advertisement for the stored data representation language representation.

Graham discloses "[a] service provider protocol adapter servlet [that] listens for service advertising requests...The service provider protocol adapter servlet then converts the service advertisement from a service protocol into a canonical representation of service advertising and stores the advertisement in an internal registry. A client protocol adapter servlet listens for client lookup requests and looks up a corresponding service provider in the internal registry...The client protocol adapter servlet then converts a client request into a canonical representation of the request, which is then used to look up the service required by the client. Once a match has been found, the client protocol adapter servlet brokers the mechanism of client-service provider interaction." (Graham, abstract). Thus, Graham teaches a "service provider protocol adapter servlet" that listens for and intercepts service advertising requests, converts service advertisements from a service protocol into a canonical representation, and stores the canonical representations in an internal registry.

Nowhere does Graham teach or suggest *converting* a current computation state of a process into a *data representation language representation* of the current computation state, nor does Graham teach or suggest *storing* the data representation language representation of the current computation state of the process. In addition, nowhere does Graham teach or suggest *generating* an advertisement for the stored data representation language representation.

Not only does the cited art (Buckle, Jagannathan and Graham) not teach or suggest converting a current computation state of a process into a data representation language representation of the current computation state, storing the data representation language representation of the current computation state of the process, and generating an advertisement for the stored data representation language representation, there is no teaching, suggestion, or motivation to combine the art to teach these limitations. Buckle teaches that the byte code of an agent, including objects and object states, may be included in a message for transmission from a location L1 to a location L2, and then may be stored locally on L2. There is no motivation within Buckle to convert the current computation state of a process into a data representation language representation of the current computation state or to store the data representation language representation of the current computation state independent of transmitting the data representation language representation to another location. Jagannathan teaches the migration of agents from one machine to another machine, even during process execution, using serialization. There is no motivation within Jagannathan to convert the current computation state of a process into a data representation language representation of the current computation state or to store the data representation language representation of the current computation state. Graham teaches a "service provider protocol adapter servlet" that listens for and intercepts service advertising requests, converts service advertisements from a service protocol into a canonical representation, and stores the canonical representations in an internal registry. There is no motivation within Graham to convert the current computation state of a process into a data representation language representation of the current computation state, to store the data representation language representation of the current computation state, or to generate an advertisement of the stored data representation language representation.

Therefore, the rejection of claim 41 is not supported by the teachings of the cited art and withdrawal thereof is respectfully requested.

Applicants also assert that the rejections of numerous ones of the dependent claims are further unsupported by the cited art. However, since the rejections of each of

the independent claims have been shown to be improper, a further discussion of the rejections of the dependent claims is not necessary at this time.

Applicants note that claims 30-32 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Buckle in view of Jagannathan in further view of the "Official Notice". However, the Examiner never stated of what he was taking "Official Notice." Therefore, the rejection is improper. Furthermore, to the extent the Examiner is taking Official Notice that it was well known in the prior art for a third device configured to retrieve the stored data representation language representation of the current computation state of the process from the second device, reconstitute the process at the current computation state from the data representation language representation of the current computation state of the process, and resume execution of the process from the current computation state, Applicants traverse. Such features were not known in the prior art and none of the references cited by the Examiner suggest these features.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-47200/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees (   ).

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: April 29, 2004